HAL
open science

Topology Construction in RPL Networks over Beacon-Enabled 802.15.4

Malisa Vucinic, Gabriele Romaniello, Laurene Guelorget, Bernard Tourancheau, Franck Rousseau, Olivier Alphand, Andrzej Duda, Laurent Damon

To cite this version:

Malisa Vucinic, Gabriele Romaniello, Laurene Guelorget, Bernard Tourancheau, Franck Rousseau, et al.. Topology Construction in RPL Networks over Beacon-Enabled 802.15.4. Computers and Communication (ISCC), 2014 IEEE Symposium on, Jun 2014, Madeira, Portugal. 10.1109/ISCC.2014.6912563 . hal-00985933v2

HAL Id: hal-00985933

https://inria.hal.science/hal-00985933v2

Submitted on 18 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Topology Construction in RPL Networks over Beacon-Enabled 802.15.4

Mališa Vučinić*¶, Gabriele Romaniello*¶, Laurène Guelorget*, Bernard Tourancheau*,
Franck Rousseau*, Olivier Alphand*, Andrzej Duda*, and Laurent Damon¶
*Grenoble Alps University, CNRS Grenoble Informatics Laboratory UMR 5217, France
¶STMicroelectronics, Crolles, France
Email: {firstname.lastname}@imag.fr, {firstname.lastname}@st.com

***Abstract*—In this paper, we propose a new scheme that allows coupling *beacon-enabled* IEEE 802.15.4 with the RPL routing protocol while keeping full compliance with both standards. We provide a means for RPL to pass the routing information to Layer 2 before the 802.15.4 topology is created by encapsulating RPL DIO messages in beacon frames. The scheme takes advantage of 802.15.4 command frames to solicit RPL DIO messages. The effect of the command frames is to reset the Trickle timer that governs sending DIO messages. We provide a detailed analysis of the overhead incurred by the proposed scheme to understand topology construction costs. We have evaluated the scheme using Contiki and the instruction-level Cooja simulator and compared our results against the most common scheme used for dissemination of the upper-layer information in *beacon-enabled* PANs. The results show energy savings during the topology construction phase and in the steady state.**

***Index Terms*—IEEE 802.15.4, beacon-enabled mode, RPL, Wireless Sensor Networks, topology construction, multi-hop networks.**

I. Introduction

The long awaited Internet of Things (IoT) has never been closer. The industry has fully begun to take part and the further development is all about standard compliance. The upper layers of the IP protocol stack for Wireless Sensor Networks (WSN) are being fine-shaped and the gaps between IETF and IEEE standards [1] are being bridged. Full solutions begin to emerge so product interoperability and security are of primary concern [2].

The IEEE 802.15.4 standard [1] is widely recognized as the main technology for low-power wireless sensor networking [3]. Among its different modes, we focus on the *beacon-enabled* mode able to achieve very low energy consumption by supporting a desired level of radio duty cycling (the proportion between the periods nodes are on and off). In this paper, we address the problem of running the Routing Protocol for Low Power and Lossy Networks (RPL) [4], the IETF standard for routing in WSN, on top of IEEE 802.15.4 *beacon-enabled* nodes.

The forwarding structure built by RPL is a Destination Oriented Directed Acyclic Graph (DODAG). Each node keeps a list of available parent nodes closer to the DODAG root (sink node) and selects one of them as the "preferred parent" based on an objective function. When a link to the preferred parent fails, a node switches to another parent in its list. At the link layer, the *beacon-enabled* IEEE 802.15.4 nodes need to construct a cluster-tree anchored at the PAN coordinator (also the sink node) for supporting multi-hop communication. Moreover, a node joining the cluster-tree has to associate with a coordinator (a Layer 2 operation) before it may send any data frame. The choice of the coordinator influences any possible choice of the RPL parent node. In the case of the *beacon-enabled* IEEE 802.15.4 nodes, the problem is how to construct the 802.15.4 cluster-tree according to the RPL routing information based on a DODAG.

While both *beacon-enabled* IEEE 802.15.4 and RPL have been extensively studied within their OSI abstraction layer, the joint operation is surprisingly still an open problem. The existing work in the literature [5] requires extensive modifications to both standards, which is an unrealistic requirement at the current stage of IoT stack development.

We propose a solution to the problem that satisfies the constraint of keeping RPL and IEEE 802.15.4 unchanged. In our approach, RPL constructs its DODAG before the cluster-tree at L2 and we use the RPL routing information (selection of the preferred parent) in the association decision to establish links, i.e., to select the coordinator in the cluster-tree that is the preferred parent in the DODAG.

The proposed solution takes advantage of cross-layer signaling: a node joining the network requests RPL information from neighbor 802.15.4 coordinators and associates with the right coordinator based on the information in the RPL DIO message (DODAG Information Object). We adapt the operation of the Trickle timer [6] that governs the transmission of DIO messages to provide the required information to Layer 2 (the adaptation remains compliant with the RPL specification).

The main contributions of the paper are the following:

- a new scheme that allows RPL to run over the *beacon-enabled* IEEE 802.15.4 without any modification to the two standards,
- the scheme leading to energy savings both during the topology construction and in the steady-state, due to the use of the Trickle timer,
- a simple probabilistic model of the Trickle timer and an analysis of the delay of the proposed scheme,
- an evaluation of energy savings and the time for topology convergence based on the implementation of the proposed scheme in Contiki.

The paper is organized as follows. We present the back-

ground information on the *beacon-enabled* IEEE 802.15.4 and RPL in Sections II and III. We provide a detailed description of the proposed scheme in Section IV and evaluate it in Section V. Section VI summarizes the related work. We conclude and discuss the future work in Section VII.

II. Beacon-enabled IEEE 802.15.4

The lack of a radio duty cycling scheme in the IEEE 802.15.4 *non-beacon* mode has led to intensive research on the *beacon-enabled* mode. Periodic beacon frames allow the synchronous sleep schedule of devices in the network. As a consequence, it is possible to *guarantee* a desired level of radio duty cycling (RDC), which is of the utmost importance for battery operated nodes and especially for those that harvest energy from the environment.

The operation of nodes in the *beacon-enabled* mode relies on beacons that delimit the start of a *superframe*. Immediately following is the Contention Active Period (CAP) during which nodes transmit pending data frames to their parent (cluster coordinator) using the slotted CSMA/CA algorithm (a coordinator node needs to stay active during CAP). Beacon Order (BO) and Superframe Order (SO) are the key parameters to tune the desired level of radio duty cycling in the *beacon-enabled* mode. Beacon Interval (BI) is defined as $BI = aBaseSuperFrameDuration * 2^{BO}$ and $SD = aBaseSuperFrameDuration * 2^{SO}$ is the CAP duration. Thus, the upper-bound proportion of time a duty cycled coordinator node will be active is 2^{SO-BO}. Leaf nodes that only wake up for a transmission may benefit from an even lower duty cycle.

The network formed in the *non-beacon* mode may be a mesh in which each node may communicate with its radio-range neighbors, so running RPL in this case does not raise any problems. Nodes in the *beacon-enabled* mode have to form a *cluster-tree*: a node selects one parent node, the cluster coordinator, and synchronizes with its beacons. The node may become a coordinator itself on behalf of other nodes, which enables multi-hop communication from leaf nodes to the root of the cluster-tree. Scheduling of active periods of different coordinators is not defined in the IEEE 802.15.4 standard and is therefore left as an implementation choice. In our evaluations, we use a simple static allocation of active periods that allows us to focus on the topology construction without affecting the overall results.

The Personal Area Network (PAN) coordinator is the root of the tree, the sink of the sensor network. It starts the topology construction by transmitting the first beacon. Other nodes are unassociated and have to switch their radio transceivers on to perform passive scanning, the only mechanism for discovering potential coordinators available in the *beacon-enabled* mode. The reception of a beacon initiates a scan period during which a node waits for beacons. At the end of this period, a node can initiate the association with the best coordinator with the sequence of `association-request`, `ack`, `data-request`, `association-reply`, `ack` control frames.

Note that most of the energy consumed during the topology construction phase comes from idle listening during the scan period, which is unavoidable for any association strategy that discovers the best available coordinator. The duration of this interval should allow the discovery of all coordinators in the radio range.

Fig. 1 illustrates a timeline of the topology construction for an example cluster-tree composed of four nodes. Note that Node 4 may receive beacons from Coordinators 2 and 3, but it selects Node 3 as the best parent.

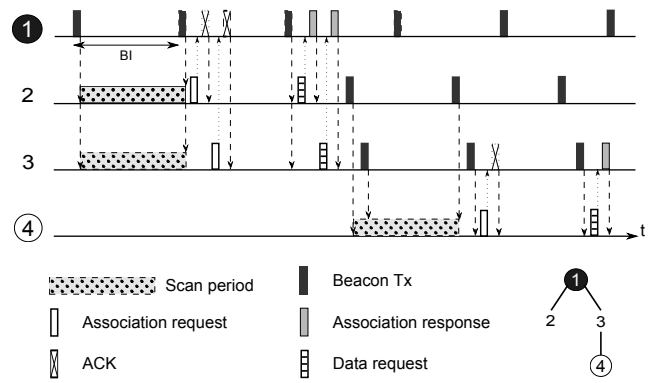


Fig. 1: Topology construction in an example 802.15.4 cluster-tree.

III. RPL—Routing Protocol for Low Power and Lossy Networks

Operating at the Network layer, RPL [4] is a Distance Vector protocol that specifies how to construct a Destination Oriented Directed Acyclic Graph (DODAG) with a defined objective function and a set of metrics and constraints.

RPL specifies a set of new ICMPv6 control messages to exchange information related to a DODAG and notably:

- *DODAG Information Object (DIO)* defines and maintains upward routes to the root, i.e. the DODAG.
- *DODAG Information Solicitation (DIS)* messages request the DODAG related information from neighboring nodes.

The root starts the DODAG building process by transmitting a DIO. Neighboring nodes process DIOs and make a decision on joining the DODAG based on the objective function and/or local policy. A node computes its *Rank* with respect to the root and starts advertising DIO messages to its neighbors with the updated information. As the process continues, each node in the network receives one or more DIO messages and selects a preferred parent towards the root. Note that for robustness, RPL keeps a list of other parents that can be used in case link conditions change. As we focus on topology construction, we limit the discussion to upward routes.

In case of *beacon-enabled* IEEE 802.15.4 at L2, the traditional layer-independent operation would confine the selection of RPL routes to those in the already-constructed L2 cluster-tree. Consequently, the overall performance of RPL would be significantly degraded. We exploit the approach of merging two structures: the 802.15.4 cluster-tree and the DODAG of

RPL, which allows us to benefit from low overhead, small delays, and near optimal upward routes of RPL [7] while creating the IEEE 802.15.4 cluster-tree required for low duty cycle communications.

RPL uses the Trickle algorithm [6] to govern the transmission interval of DIOs. The Trickle algorithm has three main parameters: i) the minimum interval size I_{min}, ii) the maximum interval size I_{max} expressed as the number of times the timer may double, iii) the redundancy constant k.

The main idea of the algorithm is to exponentially reduce the amount of control traffic in the network when the topology is consistent, i.e. when there are no link failures or arriving nodes. Consistency is checked by comparing the DODAG state advertised by other nodes in DIOs with the local one. If the number of consistent DIO receptions is higher than redundancy constant k, Trickle refrains from transmitting. Instant t at which Trickle decides if it is going to transmit is randomly selected from interval [I/2, I), where $I \in \{I_{min} \times 2^n \mid n \in Z_{\geq 0}, n \leq I_{max}\}$. Interval I is doubled upon its expiration by incrementing n. When a node detects inconsistency (which also includes the initial DODAG construction), n becomes 0, which sets interval I to I_{min}. As detailed in the next section, we suitably adapt parameter I_{min} to construct IEEE 802.15.4 cluster-tree according to the DODAG of RPL.

IV. 802.15.4 Cluster-Tree Construction Based on RPL DODAG

We propose the selection of the best coordinator in the 802.15.4 cluster-tree based on the preferred parent in the DODAG of RPL. The resulting cluster-tree will effectively be a subset of the DODAG initialized during the topology construction phase. There are several issues with such an approach:

1) RPL is a network layer protocol, but no communication among nodes at Layer 3 may take place before links at Layer 2 are established (node association with a coordinator).
2) An IEEE 802.15.4 node once associated can only communicate with its cluster coordinator, so after association, a node can only receive DIO messages from its cluster coordinator.

In this paper, we focus on addressing the first issue to enable topology construction. The second issue is a part of the future work that may add robustness to IEEE 802.15.4 cluster-tree by keeping alternate parents from the DODAG of RPL.

To address the first issue, we exploit the fact that DIO messages are multicast. As Layer 3 multicasts translate to Layer 2 broadcasts, we use beacons to broadcast DIO messages. There is no better broadcast mechanism in multi-hop *beacon-enabled* networks than the beacons themselves—during the scan period devices wait for beacons. We assume that IEEE 802.15.4 Reduced Function Devices (RFD) are configured as RPL leaf nodes, i.e., they do not send DIO messages. Similarly, Full Function Devices (FFD) may become cluster coordinators, i.e., they have to be configured as RPL routers, which is a realistic assumption as the role of a device mainly depends on its energy source. We assume that a node a priori knows if it is an RFD or an FFD.

We propose the encapsulation of RPL DIO messages in the beacon frame payload following an idea discussed in the team [8]. Layer 2 adds DIO to the payload of the next scheduled beacon if the resulting frame does not exceed IEEE 802.15.4 maximum physical layer frame size of 127 bytes (cf. Fig. 2). In case the DIO message cannot fit into the current beacon, it may be fragmented or delayed for the following one as the beacon payload size varies as a function of downward traffic.

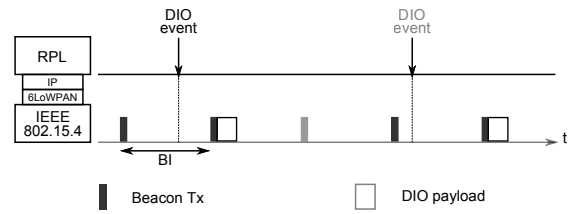


Fig. 2: Encapsulation of DIO messages in beacon frames.

The exponential increase of the DIO transmission interval governed by Trickle has an important side effect: arriving nodes would potentially wait a long time interval before receiving the first DIO. RPL addresses this issue with DIS messages that can be broadcast to solicit the transmission of a DIO: upon reception of a DIS, a node resets its Trickle interval I to I_{min} so DIO will be transmitted shortly [4]. However, DIS broadcast is not enough for synchronous duty cycled networks—neighbor nodes in the radio range may sleep at the instant of the DIS transmission. As explained above, the reception of a beacon delimits the start of the Contention Active Period during which the coordinator is active. Thus, CAP is the most suitable period during which an unassociated node may solicit information from nearby coordinators. Note that a node wanting to join the network is awake during the scanning period so it can receive beacons from several neighbor coordinators. Thus, we propose that the node transmits a solicitation message by performing CSMA/CA after the beacon if the following two conditions hold:

- the received beacon is the first one received from a given coordinator,
- the beacon does not contain a DIO in its payload.

The solicitation message could be a RPL DIS message encapsulated in an 802.15.4 command frame. Note that a node cannot send data frames before association [1]. However, we have chosen to use the IEEE 802.15.4 *beacon-request* command frame without any payload as a solicitation message—it has a small size (8 bytes) so a very short transmission time. Additionally, the RPL specification [4] allows the Trickle reset triggered by external events.

Note that the *beacon-request* command frame is typically used in the *non-beacon* mode to solicit the information about the network. It has no use in the *beacon-enabled* mode as beacons are periodically transmitted. We use its reception at Layer 2 to trigger the reset of the Trickle timer at the RPL layer

to spawn a DIO transmission. The goal is to encapsulate the DIO message in the following beacon so that arriving nodes can select the best coordinator. As a node may send several *beacon-request* solicitation frames during the scan period (and CAP of each detected coordinator), the scheme ensures the reset of the Trickle timer for all RPL routers in the range.

A possible drawback of the scheme could be its possible side effect on the duration of the always-on scan period. In fact, with typical parent selection schemes at Layer 2, each beacon carries a network-specific metric processed by arriving nodes. Then, in case Beacon Order is a priori known, the worst-case scan duration is one Beacon Interval. However, a simple algorithm achieves the same duration with our scheme as well—during the scan period of duration BI:

1) for each discovered coordinator, a node stores the expected instant of the next beacon ($current_time() + BI$),
2) for each discovered coordinator, a node solicits the reset of the Trickle timer as explained above,
3) upon expiration of BI, a node goes to sleep and schedules its wake up at the instants found in (1),
4) a node wakes up and receives the beacon with the DIO payload,
5) upon reception of the DIO payload from the last discovered coordinator, a node consults RPL about the best choice and schedules the next wake up just before the beacon of the selected coordinator; then, the node follows the standard association procedure.

This scheme ensures the discovery of all coordinators in the radio range while allowing a node to start duty cycling after one BI from the boot time (cf. Fig. 3). During next BI, node receives DIOs and passes them to RPL. In the worst case, by the end of the second BI, RPL will have the preferred parent selected. The additional *worst-case* delay of one BI is the price to pay during the topology construction for the benefit that comes later-on with the Trickle timer during the network operation. As the node spends most of the second beacon interval sleeping, it consumes energy only for receiving beacons. For n discovered coordinators, the energy will be $E = n \times T \times I_{RX} \times V$, where I_{RX} is the radio current draw in receive mode, V the operating voltage, and T transmission time of one IEEE 802.15.4 beacon with a DIO message in its payload (typically around 3.5 ms for 250 kb/s IEEE 802.15.4 compliant radios).

Note, however, that in many deployments, BO is not a-priori known. In such cases, devices have to scan for longer periods to account for the largest expected BI in presence of multiple PANs [9]. Our scheme in such scenarios introduces no additional delay as long as the preconfigured scan duration is greater than or equal to half the actual BI in the network.

A. I_{min} Parameter Tuning and Analysis

The successful operation of the proposed scheme requires that, upon solicitation, the subsequent beacon includes a DIO message. To achieve such behavior while keeping the operation of two layers independent, we need to configure

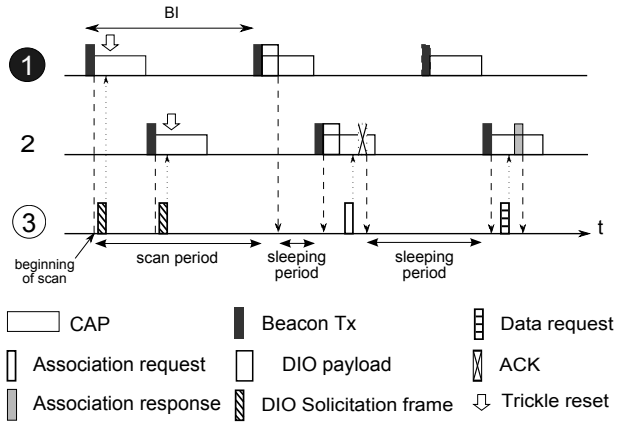


Fig. 3: Soliciting DIO during the scan period.

the Trickle I_{min} parameter as a function of BI, because the reception of a solicitation frame triggers the Trickle timer reset and the next timer value will be uniformly drawn from the interval [I_{min}/2, I_{min}). Thus, to ensure the arrival of the next DIO before the subsequent beacon, the following condition needs to hold:

$$I_{min} \leq BI - SD, \tag{1}$$

where SD denotes CAP duration. Similarly, as previously discussed, the worst case scan period when BO is a priori known, is BI. The optimal performance of Trickle with our scheme is obtained when $I_{min} = BI - SD$, which ensures the successful operation while having the lowest overhead.

B. Analysis of DIO Reception Delay

We evaluate here the expected delay of DIO messages encapsulated in periodic beacons. We define the Trickle timer value as random variable X uniformly distributed in [I/2, I), where I is a random variable denoting the current Trickle state. Then, from the Layer 2 point of view, a DIO message arrives during a beacon interval at instant $X \ mod \ BI$. Delay D is the interval remaining until the transmission of the next beacon:

$$D = BI - (X - \left\lfloor \frac{X}{BI} \right\rfloor * BI). \tag{2}$$

The expected delay is then:

$$E[D] = BI - E[X] + E[\left\lfloor \frac{X}{BI} \right\rfloor] * BI. \tag{3}$$

Now, recall that I is a discrete random variable in $\{I_{min} \times 2^n\}$, where $n = 0, 1, \ldots, I_{max}$. We model I with a discrete-time Markov chain shown in Fig. 4, where p denotes the probability of the Trickle reset. We can notice from

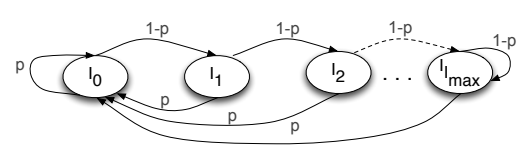


Fig. 4: Markov chain with I_{max}+1 states for Trickle.

Fig. 4 that stationary probabilities of states $I_0, \ldots, I_{I_{max}-1}$

follow a geometric distribution with reset probability p: $\Pi_{I_i} = (1-p)^i p, i = 0, \ldots, I_{max}-1$. The last state, $I_{I_{max}}$ has the stationary probability: $\Pi_{I_{I_{max}}} = (1-p)^{I_{max}}$. We can find the expected Trickle timer value as $E[X] = E[E[X|I]]$.

As our scheme uses the *beacon-request* solicitation frame at L2 to reset Trickle, the case $I = I_{min}$ is of a particular interest. From Eq. 3, it follows that:

$$E[D|_{I=I_{min}}] = BI - E[X|_{I=I_{min}}] + E[\left\lfloor \frac{X}{BI} \right\rfloor |_{I=I_{min}}] * BI. \tag{4}$$

Given the condition of Eq. 1 and also the fact that the right endpoint is excluded from the uniform interval, term $E[\lfloor \frac{X}{BI} \rfloor]$ goes to zero leaving: $E[D \mid I = I_{min}] = BI - E[X \mid I = I_{min}]$. Finally, as X is now a uniform random variable in $[I_{min}/2, I_{min})$, the expected DIO delay becomes:

$$E[D|I = I_{min}] = BI - \frac{3}{4} I_{min}, \quad I_{min} \leq BI. \tag{5}$$

We have validated Eq. 5 by emulating a real node running the Contiki operating system for constrained devices. We have timestamped the expiration instants of Trickle and the instants of the beacon with DIO transmission. We have configured I_{min} to an approximate value of $BI/2$ (Contiki accepts the values of I_{min} in power of 2). The emulation results over 5000 samples strongly corroborate our analysis with a maximal error of 2.799%.

From Eqs. 1 and 5, it follows that for setting $I_{min} = BI - SD$, our scheme introduces the least additional delay to Trickle after reset, while ensuring successful operation.

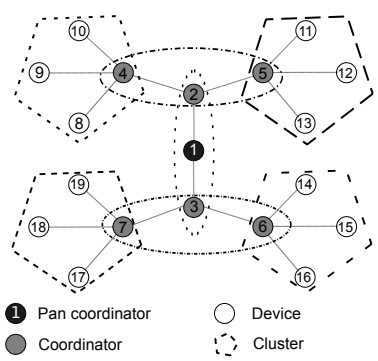


Fig. 5: Evaluated topology.

V. Performance Evaluation

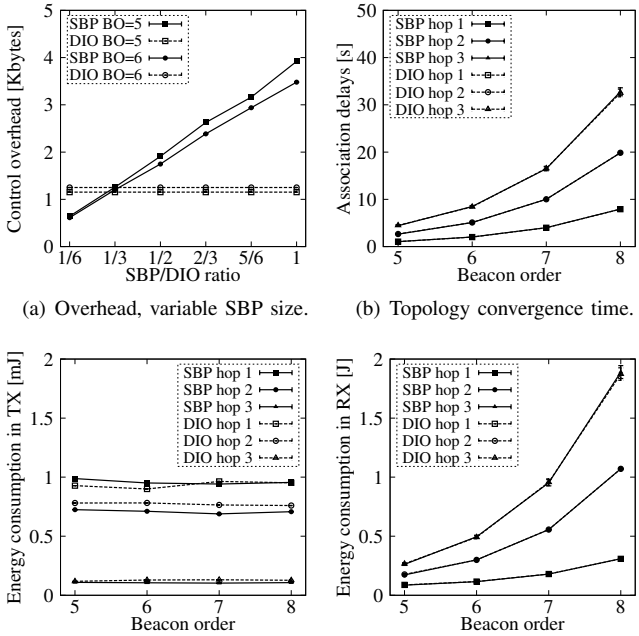
To evaluate our scheme, we have used an implementation of the IEEE 802.15.4 *beacon-enabled* mode specifically developed and optimized for harvested sensor motes manufactured by STMicroelectronics (ST) containing a 32 bit microcontroller and a proprietary radio 802.15.4 transceiver. To our knowledge, it is the first IEEE 802.15.4 *beacon-enabled* implementation for the Contiki operating system. To benefit from the Cooja simulator [10] that uses the MSPsim instruction-level emulator of the Tmote Sky platform, our team ported the *beacon-enabled* layer developed for the ST motes to the Tmote Sky platform. Tmote Sky is based on a 16-bit MSP430 microcontroller, operating at 8MHz clock rate, and the CC2420 Chipcon radio. Note that the only imperfection of Cooja with respect to the real world environment comes from the Unit Disk Graph radio channel model. Fig. 5 presents the evaluated topology.

Many authors in the literature discussed the method of encapsulating information necessary for topology construction in the beacon payload (parent selection, neighbor discovery) [11], [12], [13]. Consequently, they assume the information to be present in each beacon. As our goal in this paper was to present benefits in terms of 802.15.4 topology construction, we have compared our scheme against this approach and denote the scheme Systematic Beacon Payload (SBP). To be fair and not to loose the generality of our results, we have studied the effects of varying the SBP message size and how it affects performance. We found that the two schemes have similar performance when the SBP message size is approximately 1/3 of the DIO size (cf. Fig. 6(a)), that is, when one coordinator from Fig. 5 sends 1 DIO message for every 3 beacons with SBP on the average during topology construction. Note that this ratio depends on the duration of the scan period and the configuration of Trickle. For a given implementation, one can easily evaluate such a ratio and derive the gain or loss depending on the message size parameters.

We set the I_{min} Trickle parameter to approximately $BI - SD$ and keep SO equal to 2. We compute the radio energy consumption from the current draw values reported in the Tmote Sky data sheet. We average all the points in the following graphs over 20 emulation runs and show them with 95% confidence intervals.

We can notice in Fig. 5 that nodes have only one coordinator in their radio range. We have chosen such topology to focus on topology construction in RPL networks over *beacon-enabled* 802.15.4 and evaluate the effect of our scheme. In this way, we isolate topology construction aspects from the problems related to routing that may depend on the choice of routing metrics or objective functions. Moreover, a single coordinator discovered during the scan period BI means that the solicitation scheme is put under stress. Indeed, if a single DIO message does not arrive with the subsequent beacon upon solicitation, the node will have to initiate another scan period, which would unnecessarily increase the topology convergence delay. Nevertheless, the example topology in Fig. 5 is favorable to the proposed scheme in terms of delay—it does not introduce additional delay in case BO is a priori known, i.e., the first discovered coordinator is also the last one, so a node can initiate the association procedure after the scan period of one BI. However, we discuss the worst case delay in the presence of multiple coordinators in Section IV.

Also note that in some cases, the first beacon discovered during the scan period may already contain a DIO message. As the Trickle timer randomly selects its expiration interval and our scheme keeps the operation of two layers independent, it is a lucky outcome. In this case, a node does not need to solicit DIO as detailed in Section IV. However, a node still has

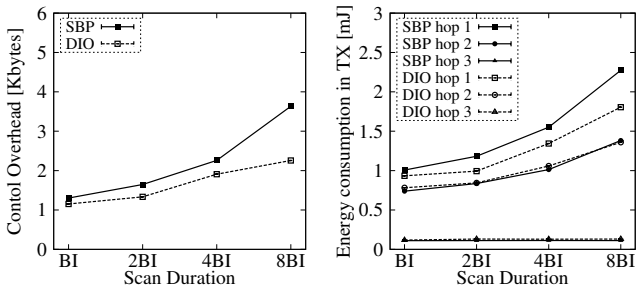


(a) Overhead, variable SBP size. (b) Topology convergence time.

(c) Energy spent in transmission, in mJ. (d) Energy spent in reception, in J.

Fig. 6: Results from emulation during topology construction.

to wait for the expiration of the scan period before initiating its association procedure to ensure that it has discovered all potential coordinators.



(a) Total overhead (b) Energy spent in transmission, in mJ

Fig. 7: Results from emulation during topology construction for variable scan duration and $BO = 5$.

We present the results for the case in which two schemes have the most similar performance, i.e., we set the message size of SBP to 1/3 of DIO (cf. Fig. 6(a)). Larger SBP message sizes result in worse performance while smaller SBP messages result in better performance during topology construction in case BO is a priori known.

A. Topology Construction

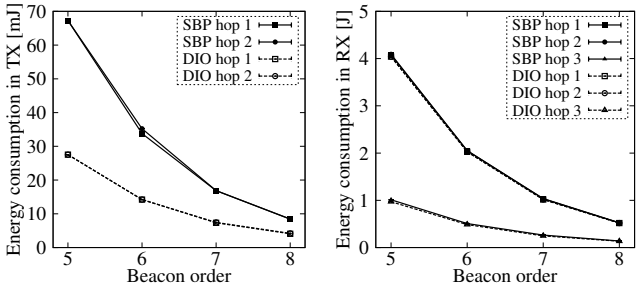
We study the topology construction phase for two cases: 1) BO is a priori known so the scan period can be set to the minimal value of BI; 2) there is no a priori knowledge of BO so nodes use a sub-optimal scan duration to account for the worst case. In both cases, simulations last until the association of the last node.

For case 1), Figs. 6(b)-6(d) present the results for varying BO. We can see in Fig. 6(b) that our scheme does not introduce any additional delay for the evaluated topology and the results for two schemes are similar within confidence intervals. Fig. 6(c) shows similar results in terms of cumulative energy spent in transmission, a consequence of the choice of the parameters for two schemes. Notably, coordinators at hop 1 and 2 spend approximately the same energy transmitting beacons. The major part of the energy spent in reception comes from idle listening during the scan period so two schemes perform equally (cf. Fig. 6(d)).

For case 2), when BO is not a priori known, we vary the scan period. As nodes remain in reception mode much longer, the energy spent in reception makes the major part of the total consumption. Similarly to Figs. 6(b) and 6(d), two schemes perform equally. However, as the scan period is longer, there is a larger number of beacons transmitted before the topology converges. We can thus see the effect of the Trickle algorithm and the proposed solicitation scheme (cf. Fig. 7(a)) that results in energy savings for hop 1 nodes as they transmit beacons the longest until the end of the tree construction (cf. Fig. 7(b)).

B. Steady-state

Furthermore, we have evaluated the benefits in terms of energy savings in the steady state, i.e., after topology construction. There was no application traffic in the network and nodes simply duty cycle according to their schedules. The presented results concern 6 minutes of the network operation after the association of the last node. We can see the effect of the reduction in control overhead by the Trickle algorithm in Fig. 8(a). In particular, FFD nodes (hop 1 and 2) transmit short beacons without any payload most of the time, which results in energy savings both during reception and transmission. During reception, however, a major part of energy consumption comes from active listening during the CAP of each coordinator so this effect is masked (cf. Fig. 8(b)). Note that in Fig. 8(a), the consumption of RFDs is zero as there is no application traffic in the network. Also, during the steady state, the reception consumption of FFDs (hop 1 and 2) is the same, as devices remain active during the same amount of time (CAP duration).



(a) Energy spent in transmission, in mJ. (b) Energy spent in reception, in J.

Fig. 8: Results from emulation during 6 min. of steady state.

VI. Related Work

The performance of multi-hop IEEE 802.15.4 networks has been well studied during the last years both using probabilistic approaches [14] and simulations [15]. Energy consumption introduced during the scan period is widely recognized as a

significant problem. The recent work of Karowski *et al.* [9] lowered this cost by optimizing the number of slots to listen over different channels. Romaniello *et al.* [16] proposed the Multichannel Beacon Train Protocol for faster discovery over multiple channels in the presence of varying beacon intervals. Kohvakka *et al.* discussed a protocol that carries the time offset and the frequency channel in beacons to ease the scanning process for the joining node [11]. It is important to stress that our work in this paper is agnostic of the scanning process. Namely, the solicitation scheme we propose starts once a node has discovered all neighboring coordinators.

As the de-facto standard for routing in IP-based WSN, RPL has been extensively studied in terms of convergence delays, route optimality, path availability, and incurred overheads [17], [18]. Coupled with the common wisdom that cross-layer signaling is necessary for a successful operation of a routing protocol in low power and lossy networks, this fact provides a strong support to the approach presented in our paper.

The work of Pavković *et al.* is closely related to ours [12]. The authors proposed the adaptations to the IEEE 802.15.4 standard to integrate its operation with RPL. Moreover, they proposed an opportunistic version of RPL to improve the delivery of time-sensitive traffic and evaluated the proposal in terms of packet delivery ratio and delay. In recent work [5], they discussed the RPL performance benefits of modifying the IEEE 802.15.4 cluster-tree structure into a "cluster-DAG". Our work was basically motivated by the same problem—the incompatibility of two structures, the 802.15.4 cluster-tree and the DODAG. While the approach of Pavković *et al.* presents performance improvement, its main drawback is the need for modifications of two standards, RPL and IEEE 802.15.4. We have addressed the same problem from a different perspective—instead of modifying the standards, we provide a means for constructing the RPL DODAG and forming the cluster-tree as its subset. As a consequence, we obtain full compliance with both standards.

VII. Conclusion

We have presented a scheme that allows coupling *beacon-enabled* IEEE 802.15.4 with the RPL routing protocol. The scheme does not require any modification to both standards. We provide a means for RPL to pass the routing information to Layer 2 before the 802.15.4 topology is created by encapsulating RPL DIO messages in beacon frames. The scheme takes advantage of 802.15.4 command frames to solicit DIO messages. The effect of the command frames is to reset the Trickle timer that governs sending of DIO messages.

We have evaluated the proposed scheme using the Contiki operating system for constrained nodes and the instruction-level Cooja simulator. The results show energy savings during the topology construction phase and in the steady state.

To isolate evaluation, we have disabled the RPL support for downward traffic (DAO messages). We plan to evaluate its behavior when running on top of the IEEE 802.15.4 cluster-tree as a part of the future work. We also plan to consider an optimal L2 mechanism that will allow the selection of an alternate coordinator from the DODAG in case of link failures.

Acknowledgments

Many thanks to Jean-Baptiste Guet and Chi-Anh La for the development of tools that we used during this work, and to Martin Heusse and anonymous reviewers for comments that improved the quality of the paper. The work of O. Alphand, F. Rousseau, and A. Duda was partially supported by the French National Research Agency (ANR) project IRIS under contract ANR-11-INFR-016 and the European Commission FP7 project CALIPSO under contract 288879. The work reflects only the authors views; the European Community is not liable for any use that may be made of the information contained herein.

References

[1] *IEEE Std 802.15.4-2011 (Revision of 2006)*, pp. 1–314, 2011.

[2] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "OSCAR: Object Security Architecture for the Internet of Things," in *WoWMoM*. IEEE, 2014.

[3] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, 2013.

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," IETF, RFC 6550, March 2012.

[5] B. Pavković, A. Duda, W.-J. Hwang, and F. Theoleyre, "Efficient Topology Construction for RPL over IEEE 802.15.4 in Wireless Sensor Networks," *Ad Hoc Networks, Elsevier*, 2013.

[6] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: a Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in *NSDI*, vol. 1. USENIX Association, 2004, pp. 2–2.

[7] A. Tripathi, J. de Oliveira, and J. Vasseur, "Performance Evaluation of Routing Protocol for Low Power and Lossy Networks," IETF, RFC 6687, 2012.

[8] O. Alphand, E. Duble, A. Duda, M. Favre, R. Guizzetti, M. Heusse, and F. Rousseau, "GreenNet: a Wireless Sensor Network for Harvested Nodes," Grenoble Informatics Laboratory, Tech. Rep., 2012.

[9] N. Karowski, A. Viana, and A. Wolisz, "Optimized Asynchronous Multichannel Discovery of IEEE 802.15.4-Based Wireless Personal Area Networks," *Transactions on Mobile Computing, IEEE*, vol. 12, no. 10, pp. 1972–1985, 2013.

[10] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Local Computer Networks*. IEEE, 2006, pp. 641 –648.

[11] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and T. D. Hämäläinen, "Energy-Efficient Neighbor Discovery Protocol for Mobile Wireless Sensor Networks," *Ad Hoc Networks, Elsevier*, vol. 7, no. 1, pp. 24 – 41, 2009.

[12] B. Pavković, F. Theoleyre, and A. Duda, "Multipath Opportunistic RPL Routing over IEEE 802.15.4," in *Proceedings of the MSWiM Conference*. ACM, 2011, pp. 179–186.

[13] Z. Yiming, Y. Xianglong, G. Xishan, Z. Mingang, and W. Liren, "A Design of Greenhouse Monitoring Control System Based on ZigBee Wireless Sensor Network," in *WiCom*, 2007, pp. 2563–2567.

[14] J. Misic, S. Shafi, and V. Misic, "Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 4, 2006.

[15] G. Anastasi, M. Conti, M. Di Francesco, and V. Neri, "Reliability and energy efficiency in multi-hop ieee 802.15.4/zigbee wireless sensor networks," in *ISCC*. IEEE, 2010, pp. 336–341.

[16] G. Romaniello, E. Potetsianakis, O. Alphand, R. Guizzetti, and A. Duda, "Fast and Energy-Efficient Topology Construction in Multi-Hop Multi-Channel 802.15.4 Networks," in *WiMob*. IEEE, 2013.

[17] M. Vučinić, B. Tourancheau, and A. Duda, "Performance Comparison of the RPL and LOADng Routing Protocols in a Home Automation Scenario," in *WCNC*. IEEE, April 2013, pp. 1974–1979.

[18] H. Kermajani and C. Gomez, "Route change latency in low-power and lossy wireless networks using rpl and 6lowpan neighbor discovery," in *ISCC*. IEEE, 2011, pp. 937–942.